

---

**From:** "Benjamin Kahn" <xkahn@cybersites.com>  
**Newsgroups:** netscape.public.mozilla.rt-messaging  
**Sent:** Thursday, May 20, 1999 9:47 PM  
**Subject:** What is Neoplanet doing?

It looks like Neoplanet is working on things internally. (At least that is what I think it looks like. They are hiring extra people, and they already have 3 programmers assigned to this project.) They may feel like waiting until they get the official go-ahead by Mozilla. But I haven't seen any other post from them.

Here is my position: My company (CyberSites, Inc.) is interested in this project. I will work with any one who is working on the project, and try to push it forward when I can. I think Michael C. Kabot is looking the right direction with his post "Starting Point."

I think it will be difficult to define people who are working on the project. In general, people move on and off OSS projects as time and interest permits. It \*is\* a good idea to find someone in charge to resolve disagreements.

We need to:

1) Define the problem. I think most of us assume the problem is already defined, but I'm not sure that's completely true. From the old mission statement:

>>

We would like to make Mozilla be able to do chatting and "instant messaging". However, we don't want to limit ourselves to a single server, or a single protocol. People today are chatting using a variety of different protocols and servers. We would like Mozilla to be able to usefully talk to all of these protocols, and hide most of the differences from the user.

On the other hand, we also don't want to hard-code knowledge of every known chat protocol into the code base. We would like to have a "pluggable" architecture, where it would be possible at any time to write up a new module for a new protocol, and Mozilla would just work with it.

<<

From the above, we have a universal chat client. The problem is to:

- \* interact with any chat protocol we can find
- \* modularize the design so new protocols are easy to add
- \* Make the use of multiple protocols transparent to the user

I would extend that slightly:

- \* Make the protocols useful at the same time
- \* Whenever possible, add security on to the protocol
- \* We should NOT define YAP. (yet another protocol)

\* The UI should "grow" to handle unusual or advanced features of a protocol or service. (This was in the original spec. You can find a copy at:  
<http://nero.cybersites.com/~xkahn/chat/notes.html>)

I'm sure there are other things we should add. It looks as if Jabber can do many of these things, and it is a good example. I wonder if the licenses are compatible?

-Ben